

In vielen Informatiksystemen werden für die Anwender Benutzerkonten angelegt, um ihnen den Zugriff auf das System zu ermöglichen, z.B. einen Zugriff auf ein Schulnetz oder auf Facebook. Während die Zugriffsberechtigungen in unserem Schulnetz eine überschaubare Anzahl darstellen, liegt die Anzahl der aktiven Nutzer bei Facebook aktuell bei 2,5 Mrd (Okt. 2020).

Die Anmeldung bei solchen Systemen benötigt einen eindeutigen Benutzernamen und ein Passwort. Möchten Sie sich anmelden, so muss der von Ihnen eingegebene Benutzername aus der Menge aller registrierten Profile herausgesucht und das Passwort überprüft werden. Erst wenn das matcht, dann wird Ihr Profil geladen. Damit dies in Sekundenbruchteilen geschehen kann, benötigt man für die Suche eine optimierte Struktur.

Aufgabe 1: Eine einfache unsortierte Liste stellt keine geeignete Struktur dar, um möglichst schnell zu einem Benutzernamen das passende Passwort herauszusuchen. Nehmen Sie zu dieser Aussage Stellung!

Wiederholung:

Ein **binärer Baum** ist

- eine Baumstruktur vom Grad zwei, d. h. jeder Knoten kann höchstens zwei Nachfolger haben.

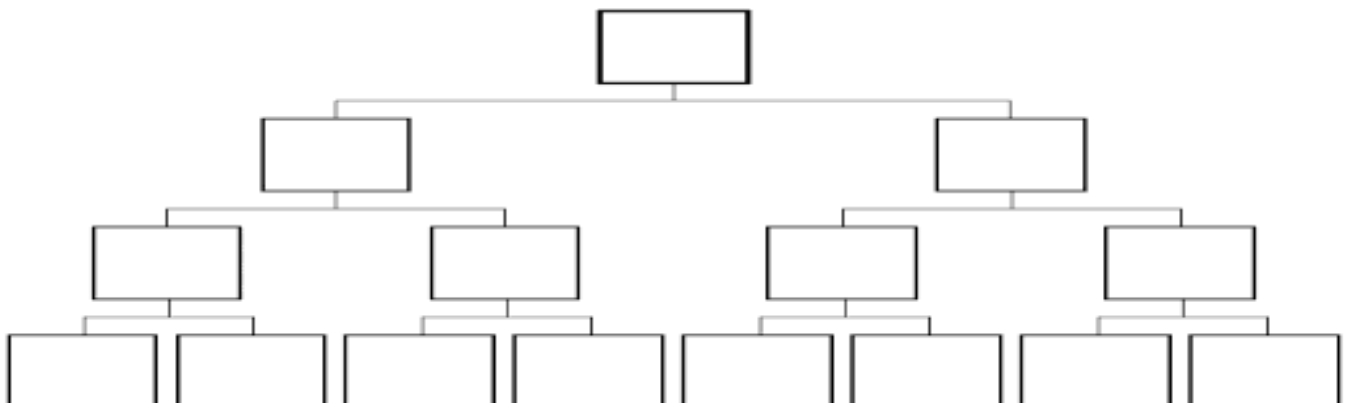
Ein **binärer Suchbaum** ist ein binärer Baum mit folgender Zusatzbedingung für alle Knoten:

- Alle kleineren (und gleichen) Schlüssel befinden sich im linken Teilbaum.
- Alle größeren Schlüssel befinden sich im rechten Teilbaum.

Das Kriterium nach dem die Daten sortiert sind, nennt man **Schlüssel**.

Aufgabe 2: Sortieren Sie Lisas Familienmitglieder nach einer lexikografischen Ordnung (wie benutzen hier den Vornamen als Schlüssel) in einen binären Suchbaum ein. Lisa bekommt den Platz in der Wurzel. Sortieren Sie dafür die Namen in der vorgegebenen Reihenfolge ein. *Wenn Sie diese Datei in einem Textverarbeitungsprogramm geöffnet haben, dann können Sie die Namen-Textfelder direkt in die Baumstruktur ziehen.*

Lisa Orville Homer Bambi Jacqueline Mona Pépé Yuma Marge Abraham J. Clancy
Simpson Simpson Simpson Petitbois Bouvier Simson Bouvier Hickman Simpson Simpson Simpson Bouvier



Aufgabe 3: Folgende Werte (Schlüssel) sind in zwei ARRAYS abgespeichert:

A1:= F(19,12,27,20,56,47,45,68,54,08,17,80,09,44,02,46);

A2:= F(11,19,2,38,17,45,3,9,81,22,47,1,83)

Konstruieren Sie daraus zwei binäre Suchbäume, wobei die erste Komponente jeweils die Wurzel ist.

Allgemeine Beschreibung der grundlegenden Operationen **Suchen**, **Einfügen**, **Löschen** von Elementen

Aufgabe 4: Beschreiben Sie ganz allgemein, das Vorgehen der Suche eines Elementes im Suchbaum.

Aufgabe 5: Beschreiben Sie ganz allgemein, wie das Einfügen von Elementen in den Suchbaum vonstatten geht.

Löschen

Auch dem Löschen geht ein Suchvorgang nach dem zu löschenden Element voraus, der das Löschen bei negativem Ausgang bereits beendet. Im anderen Fall muss das gefundene Element aus dem Baum entfernt werden. Da die Ordnung im Baum bestehen bleiben muss, unterscheidet man drei Fälle:

a) Löschen eines Blattes:

Der gefundene Knoten wird einfach entfernt.

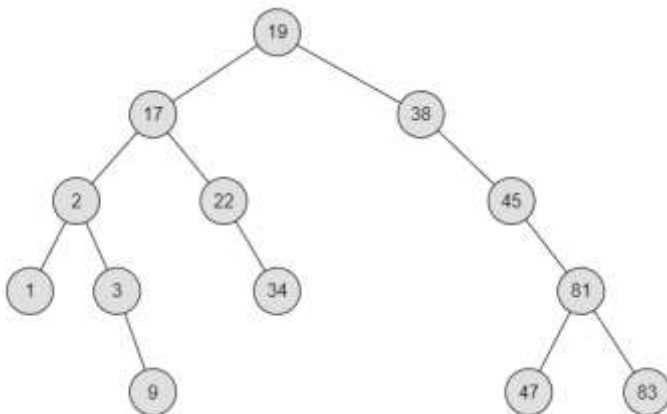
b) Löschen eines Knotens mit nur einem nicht-leeren Teilbaum:

Der gefundene Knoten wird gelöscht, und die Wurzel des nicht-leeren Teilbaumes rückt an seine Stelle. Mit der Wurzel rücken auch alle nachfolgenden Knoten, also der gesamte Teilbaum, eine Ebene höher.

c) Löschen eines Knotens mit zwei nicht-leeren Teilbäumen nach vier möglichen Methoden:

Der gefundene Knoten muss entweder *durch den größten Knoten in seinem linken Teilbaum* oder durch den kleinsten im rechten Teilbaum ersetzt werden oder der gesamte rechte (linke) Teilbaum wird an das größte (kleinste) Element des linken (rechten) Teilbaumes gehängt.

Bemerkung: programmiertechnisch entscheidet man sich vorab, welche der vier Methoden zum Einsatz kommen soll. Wenn in unserem Kurs nichts anderes vereinbart wird, dann nehmen wir die erste (kursiv geschriebene) Methode.



Aufgabe 6

Prüfen Sie bitte, ob alle Elemente richtig in den Suchbaum eingefügt wurden: B [19, 17, 38, 2, 22, 45, 1, 3, 9, 81, 47, 83, 34]. Verbessern Sie ggf.!

Aufgabe 7

Löschen Sie nun nacheinander die Elemente 34, 3, 45, 81, 17, 19 und zeichnen Sie nach den einzelnen Löschvorgängen den jeweils neu entstandenen Baum!

Aufgabe 8

Bauen Sie den Suchbaum B [5, 2, 7, 3, 1, 10, 9, 6, 11] auf. Löschen Sie anschließend in gegebener Reihenfolge die Elemente 6,5,2,1.

Aufgabe 9

Die Löschmethoden „... der gesamte rechte (linke) Teilbaum wird an das größte (kleinste) Element des linken (rechten) Teilbaumes gehängt.“ lässt sich insgesamt leichter programmieren. Warum werden diese Methoden i.d.R. aber trotzdem nicht benutzt?

Aufgabe 10

Welche Traversierung ist geeignet, um eine geordnete Liste der Elemente zu erhalten?